Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 54 (2015) 790 – 798
Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

New Data Hiding Technique In Encrypted Image: DKL Algorithm (Differing Key Length)

S. Udhayavene*, Aathira T. Dev and K. Chandrasekaran

Computer Science and Engineering, National Institute of Technology, Karnataka, Surathkal, Mangalore 575 025, India

Abstract

This paper introduces a new technique to increase the information security over the network using steganography in such a way that the secret message being sent is unidentifiable. There is a comparison made to give a clear view of how the algorithm proposed is better than LSB algorithm which is used since a long time for sending concealed messages. To avoid the chances of an attacker using steganalysis to retrieve the data, the data encryption is done. S-tool is used to show the reliability of this algorithm. We will be comparing both LSB and DKL algorithms on the basis of Mean Square Error, Peak Signal Noise Ratio, Relative Payload and Rate of Embedding. Here by its shown that DKL algorithm is more efficient than LSB algorithm.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: Encryption; Decryption; DKL; LSB; Steganography.

1. Introduction

Todays world of computer networks still has many issues in transmitting messages, keeping it secretive from any third party. Steganography is an art of concealing any kind of data, let it be text, image, audio/video, within innocuous cover carriers, which too are of the same form in a way that the secret information hidden is undetectable. In ancient times the Greek historian Herodotus was the first person to use steganography. Steganography masks the presence of any kind of communication and hence making the presence of original message not discernible to the observer. There are a few technologies which tend to be similar to steganography and these are cryptography, watermarking and fingerprinting. These are essentially concerned with the security of intellectual property.

On the basis of cover object steganography may be of many types like Audio Steganography[1–21], Video Steganography, Image Steganography etc. Image Steganography is very famous due to the popularity of digital image transmission over the web. Image Steganography uses the redundancy of digital image to hide the secret data. It may be divided into two categories.They are spatial-domain methods and frequency-domain ones. The secret messages are embedded in the image pixels directly in the spatial domain. In the frequency-domain, however, the secret image is initially transformed to frequency-domain, and afterwards the messages are embedded in the transformed frequency-domain.

*Corresponding author.
E-mail address: udhayavene@gmail.com

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.093

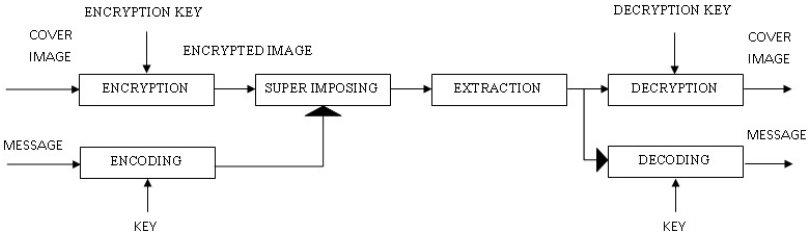

Fig. 1. Block diagram of data hider/super imposer.

Least Significant Bit (LSB) is one of the Spatial domain techniques, it uses fixed k LSBs in each pixel to embed the secret message. It is the easiest method to hide message in an image. We propose the Differing Key Algorithm (DKL) and compare its performance with LSB algorithm. Distortion is caused when pixel values are increased or decreased by one at its odd or even values. This distortion can be optimized by Optimized Pixel Adjustment Process (OPAP). Otherwise, two pixels are considered to be one for data hiding, and this method is known as Pixel Pair Matching (PPM).

Following are the factors which needs to be considered during comparison:

Secure hidden communication: The eavesdropper is kept unaware of the existence of any hidden data in the communication. The hidden data must be invisible perpetually and statistically. Steganography should produce highly imperceptible Stego-image.

Size of Payload: Unlike watermarking, which needs to embed only a small measure of copyright data, steganography goes for hidden communication and therefore usually requires sufficient embedding capacity. Requirements for higher payload and secure communication are often contradictory. A tradeoff must be looked upon based on the particular application scenarios.

Robustness: Stego-image should provide robustness to image processing techniques like compression, cropping, resizing etc., i.e. when any of these techniques are performed on stego-image, secret information should not be destroyed completely.

Many researchers through their work have found out that LSB is not optimized, on considering the above factors. It's important to keep the quality, i.e. color and dimension of the cover image undisturbed even after embedding data. It's challenging to attain high robustness to high insertion capacity at the same time.

Unlike other techniques like LSB, DKL has very less chances to get corrupted. The integrity of the hidden messages cannot be destroyed in case of DKL technique. In LSB technique, its easy for the attacker to randomize the LSB but DKL destroys all such loop holes of attack. Many research works say that more precise and accurate techniques need to be developed.

Since the key length varies for every cover image based on its pixel, the algorithm can be called "Differing Key Length (DKL)". Our work mainly focuses on comparing the LSB algorithm which is a commonly used steganographic method at every sector and DKL algorithm proposed by us. We will be comparing both the algorithms on the basis of Mean Square Error (MSE), Peak Signal Noise Ratio (PSNR), Relative Payload and Rate of Embedding. Here by its shown that DKL algorithm is more efficient than LSB algorithm.

Section 2 of our paper contains the proposed work which includes the algorithms and mathematical equations for key generation, encryption, message adding and decryption. Section 3 of our paper contains the implementation of DKL algorithm in S-tool and its comparison with LSB algorithm.

2. Related Work

2.1 Steganography

Steganography is defined as the art and science of scripting secret messages in an approach that nobody apart from sender expected recipient, guesses the presence of the message, a form of security through obscurity. The steganography word is of Greek source and signifies "concealed writing" from the Greek words steganos signifying

"protected or covered". Normally, messages will seem to be something else: articles, images, or cover-text. Classically, the secret message can be in ink which will be not visible between the seeable lines of a personal letter. This is a great security method for long data transmission.

Watermarking is a tool to conquer the frailty of present copyright laws for digitalized data. To confirm ownership and protect rights, a watermark is embedded. To save watermark from pretenders we should find the locations which are invariant to any kind of attack few of which are expansion, compression, filtering, and blurring. Each image has regions, which is called as patches, and invariant to attacks. These patches can be found by using Scale Invariant Feature Transform (SIFT) over image. Since these patches are resistant to attacks and stable, watermark is inserted in these patches. During decryption these watermarks can also be extracted successfully with low error probability. The algorithms can be strengthened using visible watermark technique.

Digitally watermarked matter will still be interoperable so that it can be seamlessly approached through heterogeneous networks and it also can be played on various playout devices that may be watermark aware or unaware. A copy attack takes place when an adversary copies a watermark from one work to another. As such, it is a way of unauthorized embedding. The copy attack attempts to thwart the capability of these systems by estimating the watermark given in an initially watermarked piece of media, and then adding that watermark to an un-watermarked piece.

Ambiguity attacks make the appearance that a watermark has been embedded in a work when in case no such embedding has come up. An adversary can use this attack to demand ownership of the distributed work. He or she may even have the capacity to make an ownership claim on the original work. As such, ambiguity attacks can be viewed as a way of unauthorized embedding. Nonetheless, they are typically considered system attacks.

The current systems use just watermarking and steganographic techniques which are prone to several attacks like Scrambling Attacks, Pathological Distortions, Copy Attacks and Ambiguity Attacks. A scrambling attack is a system-level attack in which the samples of a work are scrambled preceding to presentation to a watermark detector and afterwards subsequently descrambled. The type of scrambling can be a basic sample permutation or a more sophisticated pseudo-random scrambling of the sample values. The level of scrambling important relies on upon the detection strategy. The mosaic attack is a one in which an image is broken into numerous small rectangular patches, each too small for reliable watermark detection and it is a well-known scrambling attack.

2.2 LSB (least significant bit) technique

The LSB is the method of adjusting the carrier images LSB pixels. This technique is the simplest of all the other methods, hence vulnerable to transformations. This method was used earlier for embedding messages directly into LSB plane of an image in a deterministic order. Since the message is directly enclosed into pixel, we can lose data from the cover-image. LSB is based on the concept of that even if we change the last n, LSB for some value of n should not create intensity change to make a naked eye discover the modification.

Insertion of LSB changes based on the number of bits in the image. 24 bit bitmap image is the good image file to hide data. It is easy to conceal information when a high quality image is used. Eighth bit is used for the purpose of altering to a message bit that needs to be sent secretly in LSB method. We can store 3 bits in every pixel by altering bits of every RGB (red, green, blue) color elements for a 24 bit image. We can use 8 bits in an image which is LSB of every byte in case of BMP (gray scale). Here, secret message of 1/8th size of the image is stored.

LSB substitution is likewise workable for GIF formats, yet the issue with the GIF image is whenever the least significant bit is changed the entire colour palette will be changed. The issue can be avoided by only using the gray scale GIF images since the gray scale image contains 256 shades and the progressions will be carried out gradually so that it will be very difficult to identify. JPEG the direct substitution of steganographic strategies is impractical since it will use lossy compression. Thus it utilizes LSB substitution for embedding the data into images. There are numerous methodologies available for concealing the data within an image: one of the basic least significant bit submission approaches is "Optimum Pixel Adjustment Procedure".

Although LSB is advantageous, it has some drawbacks. It is very sensitive to filtering and manipulation of stego-image. The message might also get destroyed by cropping, scaling, noise addition, equalization of histogram, rotation and lossy compression to stego-image. It is also vulnerable to environmental noise. Picture and geometrical transformations will also destroy the hidden data. Hidden information size also depends on cover-image size and the

size of the message should be smaller than the image. Large capacity admits small cover-image for the fixed size message, hence bandwidth is reduced, which is essential in the transmission in the stego-image. A person who is attacking can also destroy the message by zeroing the whole LSB plane with a small change in perceptual feature of stego-image modified. So if a person can suspect anything hidden in the stego-image, then that steganographic method is not useful.

3. Proposed Work

First step is to find a suitable cover image to hide the data which is in text format. DKL will act as an embedding algorithm to embed the secret text into the cover image and it contains an extraction algorithm to uncover the hidden secret message.

The method of hiding the secret message in the cover image can be briefly explained as follows:

- Identifying suitable bits in the cover image $C(x, y)$.
- Prepare a suitable encryption key.
- Encrypting the secret message using the key generated.
- Superimpose one encrypted message over the other.

Each and every step is described in detail:

3.1 Key generation algorithm

1. Analyze an image and evaluate its size. Let it be $p \times q$, where p is the number of rows and q is the number of columns, of the cover image.
2. An array of size q rows and 1 column is considered, which is initialized to zero.

   $C(q) = 0$, for all q.
3. Assume the upper value of the array be $\text{value}_{\text{upper}}$ and the threshold value be $\text{value}_{\text{th}} = 0$.
4. Every new value of an array index can be calculated by:

$$\text{value}_{\text{upper}} = \sum_{x=1}^{2} \left[\prod_{i=1}^{2} \text{value}_{\text{upper}_i}(1) \right]_x \tag{1}$$

5. Everytime the array needs to be updated, it follows the following condition:

   $\text{arr(inx)} = 1$, if $\text{value}_{\text{upper}} > 0$
   $= 0$, otherwise
6. W.r.t index1 and index2 find the key of index1, where they keep changing from 1 to row size:

$$\text{key(inx1)} = 2x\,\Sigma 2\text{arr(inx1} * \text{inx2)}^{(\text{inx2}-1)} \tag{2}$$

The key generation algorithm is to develop a key and this key is generated according to the number of pixels of the cover-image. This key is further used to encrypt the message into the cover-image. This acts as an authentication code and even if a third party gets to know the stego image he cannot extract the hidden information without this key.

3.2 Encryption algorithm

1. Calculate the index value:

$$\text{inx}(i) = \text{inx}(i) - 1, \text{ where } i = 1 \text{ to } p$$
$$\text{inx}(i) = \text{inx}(i)xq$$
$$\text{key}(\text{inx}(i)) = \sum_{j=1}^{q} j + \text{inx}(i) \tag{3}$$

Message encryption using the following equation:

$$I(rq, cp) = I(rq, cq) \oplus \text{key}(\text{inx}s(i)) \tag{4}$$

where q varies from 1 to q,
p varies from 1 to p,
and i varies from 1 to 3.

Assume the message length to be maximum of 1000 characters. Test images are cover images. The message to be transmitted is encoded and cover image is encrypted using encryption key. The encoded message and encrypted image become superimposed to hide the data over a cover image.

An Encryption algorithm is used to hide the message or the secret information in such a way that only the intended recipient can extract the message. It is suitable to be used for the security in manual files on computers and storage devices because it helps to protect them from failures of physical security measures. It is also used to protect data transit i.e data transferred through internet, mobile phones, bluetooth devices etc.

3.3 Algorithm for message adding process

Maximum number of characters that can be embedded in the cover image, i.e. $c_{\max}, c_0, c_1, \ldots, c_p$. Message should be converted into binary values.

1. Concatenate each character till c_p, + refers to concatenation:

$$c_p = c_0 + \sum_{i=1}^{p} c_i \tag{5}$$

2. Consider an image to be I(R, G, B) and choose I(R).
3. Let height and length of the picture be I_h and I_l respectively,

$$I_h = I_h + \sum_{i=1}^{3} (i) \tag{6}$$

$$I_l = I_l + \sum_{j=1}^{2} (j) \tag{7}$$

4. Determine the bits where the secret message is to be hidden:

   If $k < I_l$ then $k_i = I_l \,\%\, k_i$ and $k_i = k_i + 1$
   else
   $k_i = k_i \,\%\, I_l + 1$
   if $k_j < I_h$ then $k_j = I_h \,\%\, k_j + 1$
   else
   $k_j = k_j \,\%\, I_h + 1$

   Let Array be $A(I_l, I_h)$ and initial value is equal to one

$$k_i = I_{h/2} + \sum_{j=1}^{p} k_j \tag{8}$$

$$k_j = I_{l/2} + \sum_{j=1}^{p} k_i \tag{9}$$

$c_j > (\max_c / \text{key}_l) + 2$
if A is equal to 1 then $c_j = c_j + 1$
find non zero values in an Array $A(I_l, I_h)$ and new index is $A(I_l, I_h)$.

The message adding algorithm is used for embedding the secret message inside the cover image. RGB pixels are chosen to hide the message so that there is no difference between the cover image and the resultant stego-image.

3.4 Encoding algorithm

1. Calculate inx = inx$(x_i + 2^q - 1 * (y_j - 1))$
2. Iterate 1 to $c_{\max}$ characters.
3. Iterate from 1 to $2^q - 1$ where $q = 3$
4. Let $I_l = I(R)$,
   Update $I_l(\text{inx}) = I_l(\text{inx}) + 1$, if $I(\text{inx}) \,\%\, 2 = 0$
   $I_l(\text{inx}) = I_l(\text{inx}) - 1$, otherwise

The implementation of the encoding algorithm depends on few set of lines. Differing Key Length algorithm allows the secret information to be encoded in the cover image in such a way that there is no much change in the cover image. The key is generated according to each cover image and this key acts as the stego key for each cover image. The sender encodes the cover image so that only the intended receiver can extract the concealed message.

3.5 Decoding algorithm

1. Iterate from 1 to maximum $c_{\max}$ characters.
2. Iterate from 1 to $2^q - 1$ where $q = 3$
3. Check messages (x_i, y_j) where i varies from 1 to max and j varies from 1 to $2^q - 1$; $q = 3$
4. $I_l = I(R)$ and calculate inx = inx$(x_i + 2^q - 1 * (y_j - 1))$
5. Check $I(\text{inx}) \,\%\, 2$ is equal to one, if yes then update message content by 1 w.r.t row and column.
   Row x_i varies from 1 to $\max_c$ and column y_j varies from 1 to $2^q - 1$.

4. Implementation

S-tool is the simulation used in this paper to implement the DKL algorithm. This implementation proves that along with being more efficient than LSB algorithm, DKL is one of the best ways to send secret message without any major distortion in the cover image. S-tool i.e Steganographic tool is usually used by many researchers to test various steganographic algorithm and also various steganographic techniques.

This tool supports BMP, WAV and GIF image file format. The access to the embedded algorithm is not provided to all the users, through authorization passwords the embedded algorithm was edited to test the working of DKL algorithm. The cover image needs to be dragged and put into the s-tool tab and the file containing the secret message can be dragged onto the cover image and it gets embedded in the image. Any kind of text message, audio or video files and image files can be embedded into the cover image, the only constraints that exists is the size of the secret message that can be hidden depends on the size of the cover image.

The various test images that had been chosen to implement the DKL algorithm in S-Tool and the images are given below (Fig. 2).

Sunrise Lena Soderberg Baboon Hill Zebra Boat

Fig. 2. Test images.

Figure 3 denotes the image which contains hidden text but looks same as the Original image.
The metrics used to compare LSB and DKL algorithm are as follows:

4.1 MSE calculation

The distortion in the image can be measured using Mean Square Error. MSE can be calculated using the formula given below:

$$\text{mean} = \sum_{x=1,y=1}^{p,q} (\text{Pix}_{BEx} - \text{Pix}_{AEx})(\text{Pix}_{BEy} - \text{Pix}_{AEy}/(p * q)) \tag{10}$$

Pix_{AE} = after embedding pixel values
Pix_{BE} = before embedding pixel values
$p * q$ = size of the image.

The MSE of the current LSB method and proposed DKL method are shown in Fig. 4. Here, we can observe that the proposed DKL method reduces the error. The figure shows the image modified against to visual falsifications.

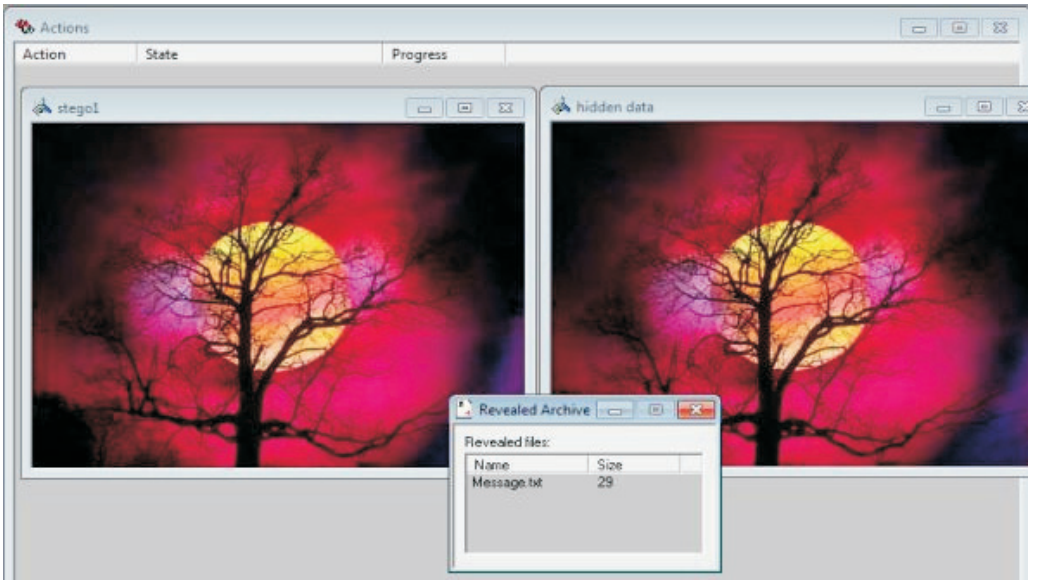

Fig. 3. Revealing the hidden data.

Table 1. Simulation result.

Name of the image	Encryption key length ($2q = p * q$)	Message length	Number of bits	LSB method MSE	LSB method PSNR in db	DKL method MSE	DKL method PSNR in db	Image Type	Original Image Size	Image Size After Encoding
Baboon	$x = 18$	18 characters	126 bits	0.0025	74.2386	0.096	34.2386	BMP	512×512	512×512
Lena										
Soderberg	$x = 18$	15 characters	105 bits	0.0023	74.2143	0.0966	34.2143	BMP	512×512	512×512
Zebra	$x = 20$	10 characters	70 bits	0.000578	80.5048	0.0958	30.5048	BMP	1024×1024	1024×1024
Hill	$x = 18$	16 characters	112 bits	0.0023	74.2514	0.0958	34.2514	BMP	512×512	512×512
Sun Rise	$x = 16$	22 characters	154 bits	0.0091	68.5334	0.0357	38.5334	BMP	256×256	256×256
Boat	$x = 16$	24 characters	168 bits	0.0090	68.5724	0.0354	38.5724	BMP	256×256	256×256

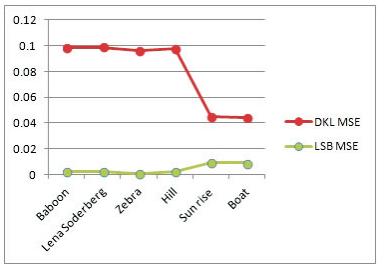

Fig. 4. Mean square error.

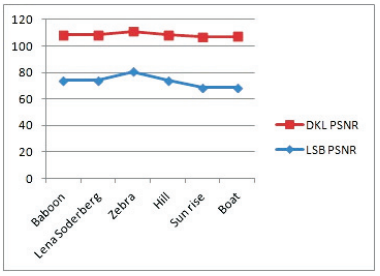

Fig. 5. Peak signal noise ratio.

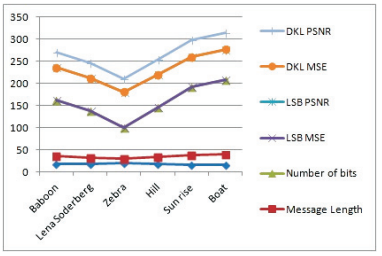

Fig. 6. MSE and PSNR over various parameters.

Table 2. Number of hidden bits and bit positions used.

Number of hidden bits	Number of bit positions used
126	63
105	52
70	35
112	56
154	67
168	84

4.2 PSNR calculation

Peak Signal Noise Ratio is the ratio of peak square value of pixels by MSE and is expressed in decibel. It is used for the measurement of the mathematical difference among the stego-image and cover image.

$$PSNR = 10\log_{10}(2^q - 1/MSE) \tag{11}$$

where q depends on the number of bits to represent pixel of an image.

The PSNR of existing and proposed algorithm is shown in Fig. 5 which shows a significant improvement of PSNR comparing with the current LSB scheme.

4.3 Relative payload (RPl)

RPl is the hidden message length or the no of positions used for modification. Let m_l be the message length and the number of positions be pos_n used for modification.

$$RPl = m_l/pos_n \tag{12}$$

4.4 Rate of embedding

Let E_{effect} be the number of hidden message per modification and modified bits be bit m_l and msg_h be the number of hidden message. Graph of LSB and DFL algorithms was drawn considering various parameters like the number of bits, the length of the message, MSE and PSNR.

5. Conclusion

This paper explores deep into the proposed technique of using DKL algorithm for data encryption and transmission to be more efficient than the earlier used technique of using an LSB algorithm for the same. The comparison proves DKL is better than LSB. S-tool has been used to implement the DKL algorithm which successfully encrypts the data into an image which can then be sent as a normal image file to the appropriate recipient. The future work of this paper proceeds to implementation of DKL algorithm in MATLAB.

References

[1] Eric Cole and Ronald D. Krutz, Hiding in Plain Sight: Steganography and the Art of Covert Communication, Wiley Publishing Inc, (2003).
[2] David Kahn, The History of Steganography, *Proc. of First Int. Workshop on Information Hiding*.
[3] Artz, D, Digital Steganography: Hiding data within Data, *IEEE Internet Computing*, May/June (2001).
[4] R. Crandall, Some Notes on Steganography, Posted on Steganography Mailing List, (1998) [Online]. Available: http://os.inf.tu-resden.de/westfeld/crandall.pdf
[5] M. Hossain, S. A. Haque and F. Sharmin, Variable Rate Steganography in Gray Scale Digital Images Using Neighborhood Pixel Information, *The International Arab Journal of Information Technology*, vol. 7, no. 1, pp. 34–38, (2010).
[6] R. J. Anderson, Stretching the Limits of Steganography, *Springer Lecture Notes in Computer Science*, vol. 1174, pp. 39–48, (1996).
[7] N. J. Hopper, Toward a Theory of Steganography, Ph.D. Dissertation, School of Computer Science Carnegie Mellon University, Pittsburgh, PA, USA, July (2004).
[8] N. F. Johnson, Z. Duric and S. Jajodia, Information Hiding: Steganography and Watermarking Attacks and Countermeasures, *Kluwer Academic Publishers*, (2000). Available at http://www.jjtc.com/Steganography/.
[9] M. T. Chapman, Hiding the Hidden: A Software System for Concealing Ciphertext as Innocuous Text, *Master's Thesis, University of Wisconsin-Milwaukee*, May (1997).
[10] S. S. Agaian, R. C. Cherukuri and R. R. Sifuentes, Key Dependent Covert Communication System using Fibonacci p-Codes, *IEEE International Conference on System of Systems Engineering*, pp. 1–5, (2007).
[11] T. Morkel, JHP Eloff and M. S. Olivier, An Overview of Image Steganography, In *Proceeding of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, Sand to South Africa, June/July (2005).
[12] Moerland, T., Steganography and Steganalysis, Leiden Institute of Advanced Computing Science, Silman, J., Steganography and Steganalysis: An Overview, SANS Institute, 2001 Jamil, T., Steganography: The art of Hiding Information is Plain Sight, IEEE Potentials, 18:01, (1999).
[13] V. Lokeswara Reddy, A. Subramanyam, P. Chenna Reddy, Implementation of LSB Steganography and its Evaluation for Various File Formats, *Int. J. Advanced Networking and Applications 868*, vol. 2, issue 5, pp. 868–872, (2011).
[14] A. Nikolaidis, S. Tsekeridou, A. Tefas and V. Solachidi, A Survey on Watermarking Application Scenarios and Related Attacks, *IEEE International Conference on Image Processing*, vol. 3, pp. 991–993, October (2001).
[15] Deshpande Neeta and Kamalapur Snehal, Daisy Jacobs Implementation of LSB Steganography and Its Evaluation for Various Bits Digital Information Management, 2006, *1st International Conference* pp. 173–178, (2007).
[16] M. Kharrazi, H. T. Sencar and N. Memon, Performance Study of Common Image Steganography and Steganalysis Techniques, *Journal of Electronic Imaging, SPIE Proceedings*, vol. 5681.15(4), 041104, pp. 1-16, (2006).
[17] Stepfan Katzenbeisser, Fabien A. P. Patitcolas, Information Hiding Techniques for Steganography and Digital Water Mark, Chapter no. 3, pp. 56. Also available at amazon.com.
[18] K. S. Dipti and B. Neha, Proposed System for Data Hiding Using Cryptography and Steganography, *International Journal of Computer Applications*, vol. 8(9), pp. 7–10, (2010).
[19] Pedram Hayati, Vidyasagar Potdar and Elizabeth Chang, A Survey of Steganographic and Steganalytic Tools for the Digital Forensic Investigator, Institute for Advanced Studies in Basic Science of Zanjan, Iran 2 Digital Ecosystems and Business Intelligence Institute, Curtin Business School, Curtin University of Technology, Perth, Australia.
[20] R. Sridevi, A. Damodaram and S. Narasimham, Efficient Method of Audio Steganography By Modified LSB Algorithm and Strong Encryption Key with Enhanced Security, *Journal of Theoretical and Applied Information Technology*, pp. 768–771, (2009).
[21] A. Westfeld, F5A Steganographic Algorithm: High Capacity Despite Better Steganalysis, *Proc. 4th Int'l Workshop Information Hiding, Springer-Verlag*, pp. 289–302, (2001).